User's Manual of the Library MSARGIBB Version 0.1

Matteo Pelagatti

1 Introduction

This library consists of a number of procedures, which allow a quick and modular writing of the code related to the bayesian analysis of dynamic models. Although the routines are not too optimized from a programming point of view, they have proved to be reasonably fast and robust.

The choice of splitting the problems into smaller problems has permitted an easier debugging and testing of the code and a great flexibility in model building.

Once the ascii file `msargibb.src` has been placed in the *src* directory of Gauss, the library *msargibb* can be created using the syntax `lib msargibb msargibb.src`. The library can than be loaded into a Gauss session or program just with the instruction `library msargibb`, letting all the included procedures be visible to Gauss.

In the next sections, the use of every procedure in the library will be explained. All the procedures use only local variables, and no global variables are neither created, nor modified by the routines. The order in which the procedures are presented is alphabetical.

The library may contain bugs, so use it at your own risk.

2 COV_AR

- **Purpose**
  Calculates the $p \times p$ autocorrelation matrix of a stationary AR(p) process with unit innovation's variance, given the p autoregressive parameters.

- **Format**
  `Sigma = COV_AR(phi)`

- **Input**
  `phi` p-dimensional (column or row) vector of autoregressive parameters.

- **Output**
  `Sigma` $p \times p$ autocorrelation matrix of the AR(p) process.

3 ERGODIC

- **Purpose**
  Calculates the vector of ergodic (or stationary) probabilities of an ergodic Markov chain with finite state space, given the transition (or stochastic) matrix.

- **Format**
  `v = ERGODIC(P)`

- **Input**
  `P` Transition matrix (either the columns or the rows must sum to one).

- **Output**
  `v` vector of ergodic probabilities.

4 ETALIZER

- **Purpose**
  Calculates the value of the likelihood functions of an autoregressive mean-switching model for every possible state.

- **Format**
  `L = ETALIZER(y,mu,sigma,phi,S)`

- **Input**

`y`	$(n \times 1)$ vector containing the n observations.
`mu`	(2×1) vector containing the means of the switching process: mu[0] being the mean of the first state, and mu[0] + mu[1] being the mean of the second state.
`sigma2`	$(n \times 1)$ vector (in case of etheroskedasticity) of variances or scalar variance value (in case of omoschedasticity) of the model's innovations.
`phi`	$(p \times 1)$ vector of autoregressive parameters.
`S`	$(k \times p+1)$ binary (0-1) matrix of states and lagged states: column 1 contains the k states at time t, column 2 contains the k states at time $t-1$,..., column $p+1$ contains the k states at time $t-p$.

- **Output**

`L`	$(k \times n-p)$ matrix containing the the likelihood values for every period (column index $= p+1, \ldots, n$) and every state (row index $= 1, \ldots, k$). The first p columns are lost.

5 HAM_FLT

- **Purpose**

  Calculates Hamilton's filter: the filtered probabilities of being in each of the m states of a Markov switching model at every time $t = 1, \ldots, n$.

- **Format**

  `xiflt = HAM_FLT(xi0,P,eta)`

- **Input**

`xi0`	$(m \times 1)$ vector containing the initial (i.e. time $t = 0$) probabilities of the m states.
`P`	$(m \times m)$ transition matrix of the Markov chain. Important: the matrix must be stochastic in the columns, that it, the elements of every column must sum to one.
`eta`	$(m \times n)$ matrix containing the the likelihood values for every period (column index $= 1, \ldots, n$) and every state (row index $= 1, \ldots, m$)

- **Output**

`xiflt`	$(m \times n)$ matrix containing the filtered probabilities of being in state i, $i = 1, \ldots m$ row index, at time t, $t = 1, \ldots, n$ column index.

6 KIM_SMO

- **Purpose**
  Calculates Kim's smoother: the smoothed probabilities of being in each of the m states of a Markov switching model at every time $t = 1, \ldots, n$.

- **Format**
  `xismo = kim_smo(xiflt,P)`

- **Input**

  `xiflt` ($m \times n$) matrix containing the filtered probabilities of being in state i ($i = 1, \ldots m$ row index) at time t ($t = 1, \ldots, n$ column index). It can be computed using the procedure HAM_FLT.

  `P` ($m \times m$) transition matrix of the Markov chain. Important: the matrix must be stochastic in the columns, that it, the elements of every column must sum to one.

- **Output**

  `xismo` ($m \times n$) matrix containing the smoothed probabilities of being in state i, $i = 1, \ldots m$ row index, at time t, $t = 1, \ldots, n$ column index.

7 LIN_SAMP

- **Purpose**
  It generates random values from the posterior distributions of the parameters $\boldsymbol{\beta}$ and σ^2 of the following Bayesian Gaussian linear model

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$$

  with prior distributions $\sigma^2 \sim \mathcal{IG}(n_0/2, nv_0/2)$ and $\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}(\mathbf{b}_0, \sigma^2 \mathbf{A}_0)$.

- **Format**
  `{beta,sigma2} = lin_samp(y,X,b0,A0,n0,nv0)`

- **Input**

`y`	$(n \times 1)$ vector of observations.
`X`	$(n \times k)$ matrix of covariates.
`b0`	$(k \times 1)$ vector of prior means for $\boldsymbol{\beta}$.
`A0`	$(k \times k)$ matrix of prior covariances when $\sigma^2 = 1$.
`n0`	(1×1) two times the first parameter of the inverse gamma distribution (it can be interpreted as the number of observations on which the prior information about σ^2 is based).
`nv0`	(1×1) two times the second parameter of the inverse gamma distribution.

- **Output**

`beta`	$(k \times 1)$ vector of random values from the posterior of $\boldsymbol{\beta}$.
`sigma2`	(1×1) random value from the posterior of σ^2.

8 MSVARTA

- **Purpose**
  Calculates the value of the likelihood functions of a vector autoregressive mean-switching model for every possible state. It is the multivariate analogous of procedure *ETALIZER*.

- **Format**
  `eta = msvarta(Y,mu0,mu1,Sig,A,S)`

- **Input**

`Y`	$(k \times n)$ matrix containing k time series with n observations.
`mu0`	$(k \times 1)$ vector of state 0 means.
`mu1`	$(k \times 1)$ vector of state 1 means' increments: state 1 means equal `mu0 + mu1`.
`Sig`	$(k \times k)$ covariance matrix of the model's innovations.
`A`	$(k \times pk)$ matrix of VAR coefficient matrices: $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_p]$.
`S`	$(m \times p + 1)$ binary (0-1) matrix of possible states at lags $0, 1, \ldots, p$.

- **Output**

`eta`	$(m \times n - p)$ matrix of likelihood values for every state (rows) and every period (columns); the first p columns are lost.

9 NEW_BETA

- **Purpose**
  Generates a new set of probit model's coefficients $\boldsymbol{\beta}$, using the data augmentation technique and Bayesian inference.

- **Format**
  `beta1 = new_beta(s,X,beta0,diffuse,m0,V0)`

- **Input**

`s`	$(n \times 1)$ binary (0-1) vector of response variable observations.
`X`	$(n \times k)$ matrix of covariates (or design matrix).
`beta0`	$(k \times 1)$ vector of the last generated coefficients $\boldsymbol{\beta}$.
`diffuse`	(1×1) when equal to 0 a proper normal prior $\mathcal{N}(\mathbf{m}_0, \mathbf{V}_0)$ is used, otherwise a diffuse prior is adopted.
`m0`	$(k \times 1)$ mean vector of the normal prior distribution for the coefficients $\boldsymbol{\beta}$ (if `diffuse = 0`, it is not used).
`V0`	$(k \times k)$ covariance matrix of the normal prior distribution for the coefficients $\boldsymbol{\beta}$ (if `diffuse = 0`, it is not used).

- **Output**

`beta1`	$(k \times 1)$ vector of new generated values from the posterior distribution of $\boldsymbol{\beta}$.

10 RNDTN

- **Purpose**
  Auxiliary function used by *new_beta*: it generates a series of truncated normal random variables according to the rule

$$\begin{array}{ll} \text{if } s_t = 0 & \text{than } z_t \sim \mathcal{N}(\mu_t, 1)\mathbb{I}_{(-\infty,0)}(z_t) \\ \text{if } s_t = 1 & \text{than } z_t \sim \mathcal{N}(\mu_t, 1)\mathbb{I}_{[0,\infty)}(z_t) \end{array} ,$$

  $\mathbb{I}_{\{.\}}$ being the indicator function.

- **Format**
  `z = rndtn(s,mu)`

- **Input**

`s`	$(n \times 1)$ binary (0-1) vector.
`mu`	$(n \times 1)$ mean vector.

- **Output**

`z`	$(n \times 1)$ sample of truncated normal variables.

11 NEW_MU

- **Purpose**
  It generates values for the two means of an autoregressive Markov switching model in the means, MSM-AR(p), from their posterior distribution, given either Gaussian or diffuse priors, and the other parameters of the model.

- **Format**
  `mu = new_mu(y,phi,s,sigma,diffuse,m0,V0)`

- **Input**

`y`	$(n \times 1)$ observations' vector.
`phi`	$(p \times 1)$ parameters of the AR(p) model.
`s`	$(n \times 1)$ binary (0-1) state vector: when $s_t = 0$ the process has mean μ_0, when $s_t = 1$ the process has mean $\mu_0 + \mu_1$.
`sigma`	$(n \times 1)$ standard deviation of the innovations of the process (it can be a scalar).
`diffuse`	(1×1) when equal to 0 a proper normal prior $\mathcal{N}(\mathbf{m}_0, \mathbf{V}_0)$ for the means is used, otherwise a diffuse prior is adopted.
`m0`	(2×1) mean vector of the normal prior distribution for the means $\boldsymbol{\mu}$ (if `diffuse = 0`, it is not used).
`V0`	$(2 \times k)$ covariance matrix of the normal prior distribution for the means $\boldsymbol{\mu}$ (if `diffuse = 0`, it is not used).

- **Output**

`mu`	$(n \times 1)$ sample from the posterior of the means of the MSM-AR process.

12 NEW_PHI

- **Purpose**
  It generates values for the autoregressive parameters of an autoregressive Markov switching model in the means, MSM-AR(p), from their posterior, given either Gaussian or diffuse priors, and the other parameters of the model.

- **Format**
  `phi = new_phi(y,mu,s,sigma,p,diffuse,m0,V0)`

- **Input**

`y`	$(n \times 1)$ observations' vector.
`mu`	(2×1) means of the MSM-AR(p) model.
`s`	$(n \times 1)$ binary (0-1) state vector: when $s_t = 0$ the process has mean μ_0, when $s_t = 1$ the process has mean $\mu_0 + \mu_1$.
`sigma`	$(n \times 1)$ standard deviation of the innovations of the process (it can be a scalar).
`diffuse`	(1×1) when equal to 0 a proper normal prior $\mathcal{N}(\mathbf{m}_0, \mathbf{V}_0)$ for the autoregressive coefficients is used, otherwise a diffuse prior is adopted.
`m0`	$(p \times 1)$ mean vector of the normal prior distribution for the autoregressive coefficients $\boldsymbol{\phi}$ (if `diffuse = 0`, it is not used).
`V0`	$(p \times p)$ covariance matrix of the normal prior distribution for the autoregressive coefficients $\boldsymbol{\phi}$ (if `diffuse = 0`, it is not used).

- **Output**

`phi`	$(n \times 1)$ sample from the posterior of the autoregressive coefficients of the MSM-AR process.

13 NEW_SIGMA2

- **Purpose**
  It generates a random value from the posterior of the innovations' variance of an autoregressive Markov switching model in the means, MSM-AR(p), given either an inverse gamma or diffuse prior, and the other parameters of the model.

- **Format**
  `sig2 = new_sigma2(y,mu,phi,s,diffuse,alpha,beta)`

- **Input**

`y`	$(n \times 1)$ observations' vector.
`mu`	(2×1) means of the MSM-AR(p) model.
`phi`	$(p \times 1)$ vector of the p autoregressive parameters.
`s`	$(n \times 1)$ binary (0-1) state vector: when $s_t = 0$ the process has mean μ_0, when $s_t = 1$ the process has mean $\mu_0 + \mu_1$.
`diffuse`	(1×1) when equal to 0 a proper inverse gamma prior $\mathcal{N}(\mathbf{m}_0, \mathbf{V}_0)$ for the innovations' variance is used, otherwise a diffuse prior is adopted.
`alpha`	(1×1) first parameter of the inverse gamma distribution (if `diffuse = 0`, it is not used).
`beta`	$(1 \times p)$ second parameter of the inverse gamma distribution (if `diffuse = 0`, it is not used).

- **Output**

`sig2`	(1×1) sample from the posterior of the innovations' variance of the MSM-AR process.

14 NY

- **Purpose**
  It graphs a line plot of one or more series.

- **Format**
  `ny(y)`

- **Input**

`y`	$(n \times k)$ matrix containing k series of length n.

- **Output**

  no output.

15 PROBIT

- **Purpose**
  It calculates the varying probability of transition of a two states Markov chain, using the following probit duration dependence model

$$\text{probit}(S_t) = \beta_1 + \beta_2 S_{t-1} + \beta_3 D_{t-1} S_{t-1} + \beta_4 D_{t-1}(1 - S_{t-1})$$

  with S_t binary (0-1) state variable, and D_t duration variable, that counts the number of consecutive periods since last change of state, in which the system has been in the same state, at time t.

- **Format**
  `prob = probit(par,maxdur)`

- **Input**

`par`	(4×1) vector of the β-parameters of the model.
`maxdur`	(1×1) number of periods, after which the durations no more influence the probability of transition.

- **Output**

`prob`	$(maxdur \times 4)$ matrix of the following probabilities of transition after d times: $p_{00}(d), p_{01}(d), p_{10}(d), p_{11}(d)$.

16 PROBIT2

- **Purpose**
  It calculates the varying probability of transition of a two states Markov chain, using the following probit duration dependence model

  $$\text{probit}(S_t) = \beta_1 S_{t-1} + \beta_2 D_{t-1} S_{t-1} + \beta_3 (1 - S_{t-1}) + \beta_4 D_{t-1}(1 - S_{t-1})$$

  with S_t binary (0-1) state variable, and D_t duration variable, that counts the number of consecutive periods since last change of state, in which the system has been in the same state, at time t.

- **Format**
  `prob = probit2(par,maxdur)`

- **Input**

`par`	(4×1) vector of the β-parameters of the model.
`maxdur`	(1×1) number of periods, after which the durations no more influence the probability of transition.

- **Output**

`prob`	$(maxdur \times 4)$ matrix of the following probabilities of transition after d times: $p_{00}(d), p_{01}(d), p_{10}(d), p_{11}(d)$.

17 RNDDIS

- **Purpose**
  It generates n discrete random quantities with support $\{1, 2, \ldots, m\}$ and a given probability distribution.

- **Format**
  `x = rnddis(n,p)`

- **Input**

  `n` (1×1) number of random quantities to be generated.

  `p` $(m \times 1)$ vector containing the ordered probabilities for each event $\{1, 2, \ldots, m\}$. The sum of the m values must be one.

- **Output**

  `x` $(n \times 1)$ vector of n sample points of the discrete random quantities.

18 RNDIGAM

- **Purpose**

  It generates a matrix of i.i.d. random quantities from an Inverse-Gamma distribution

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{e^{-\beta/x}}{x^{\alpha+1}} \mathbb{I}_{[0,\infty)}(x),$$

  which is the distribution of the reciprocal of a Gamma random variable with parameters α and β.

- **Format**

```
x = rndigam(r,c,alpha,beta)
```

- **Input**

  `r` (1×1) number of rows of the output matrix.

  `c` (1×1) number of columns of the output matrix.

  `alpha` (1×1) α-parameter of the Inverse-Gamma distribution.

  `beta` (1×1) β-parameter of the Inverse-Gamma distribution.

- **Output**

  `x` $(r \times c)$ matrix of i.i.d. Inverse-Gamma random numbers.

19 RNDMTN

- **Purpose**

  It draws a $(p \times q)$ random matrix from a matrix-normal distribution with mean-matrix $\mathbf{M}$ rows-covariance matrix $\mathbf{V}$ and columns-covariance matrix $\mathbf{K}$.

- **Format**

```
X = rndmtn(M,V,K)
```

- **Input**
  - `M` $(p \times q)$ matrix of means.
  - `V` $(p \times p)$ rows-covariance matrix.
  - `K` $(q \times q)$ columns-covariance matrix.

- **Output**
  - `X` $(p \times q)$ matrix of random matrix-normal numbers.

20 RNDMVN

- **Purpose**
  It draws a random vector from a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

- **Format**
  `x = rndmvn(mu,Sigma)`

- **Input**
  - `mu` $(m \times 1)$ mean vector.
  - `Sigma` $(m \times m)$ covariance matrix.

- **Output**
  - `x` $(m \times 1)$ vector of multivariate normal numbers.

21 RNDW

- **Purpose**
  It draws a random $(m \times m)$ matrix from a Whishart distribution with parameters $\mathbf{H}$ and $n \in \mathbb{N}$.

- **Format**
  `X = rndw(H,n)`

- **Input**
  - `H` $(m \times m)$ parameter's matrix.
  - `n` (1×1) number of degrees of freedom.

- **Output**
  - `X` $(m \times m)$ matrix drawn from a Whishart distribution.

22 RND_MU

- **Purpose**
  It generates a vector of random quantities from the posterior distribution of the stacked means of the Markov-switching-in-mean VAR (MSM-VAR(p)) model

  $$\mathbf{y}_t = \boldsymbol{\mu}_0 + \boldsymbol{\mu}_1 s_t + \mathbf{A}_1(\mathbf{y}_{t-1} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 s_{t-1}) + \ldots + \mathbf{A}_p(\mathbf{y}_{t-p} - \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 s_{t-p}) + \boldsymbol{\varepsilon}_t,$$

  given the other parameters and a normal prior distribution.

- **Format**
  `mu = rnd_mu(Y,s,p,A,Sig,m0,invM0)`

- **Input**

`Y`	$(k \times n)$ matrix containing in the rows the k observed time series.
`s`	$(n \times 1)$ or $(1 \times n)$ binary (0-1) vector of states.
`p`	(1×1) number of lags of vector autoregression.
`A`	$(k \times kp)$ matrix of all the autoregressive parameters: $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_p)$.
`Sig`	$(k \times k)$ covariance matrix of vector innovations $\boldsymbol{\varepsilon}_t$.
`m0`	$(2k \times 1)$ prior mean vector for $(\boldsymbol{\mu}_0', \boldsymbol{\mu}_1')'$.
`invM0`	$(2k \times 2k)$ inverse of the prior covariance matrix for $(\boldsymbol{\mu}_0', \boldsymbol{\mu}_1')'$.

- **Output**

`mu`	$(2k \times 1)$ random number drawn from the posterior distribution of $(\boldsymbol{\mu}_0', \boldsymbol{\mu}_1')'$.

23 RND_REG

- **Purpose**
  It draws a random matrix from the posterior distribution of the parameter matrix $\mathbf{B}$ of the following multivariate regression model:

  $$\mathbf{Y} = \mathbf{BX} + \mathbf{E},$$

  where $\mathbf{Y}$ is a matrix of k variates with n observations, $\mathbf{X}$ is a matrix of m covariates with n observations, $\mathbf{B}$ is a $(k \times m)$ matrix of regression parameters and $\mathbf{E}$ is a matrix whose n columns are random vectors with zero expectation and covariance matrix $\boldsymbol{\Sigma}$. The prior for the regression parameters is matrix normal, while the prior for $\boldsymbol{\Sigma}$ is diffuse.

- **Format**
  {beta,Sigma} = rnd_reg(Y,X,b0,invBB0)

- **Input**

Y	$(k \times n)$ observation matrix with the n observations of the k variates.
X	$(m \times n)$ observation matrix with the n observations of the m covariates.
b0	$(km \times 1)$ vector of prior means for vec($\mathbf{B}$).
invBB0	$(km \times km)$ inverse of the prior covariance matrix for vec($\mathbf{B}$).

- **Output**

beta	$(km \times 1)$ random draw from the posterior of vec($\mathbf{B}$).
Sigma	$(k \times k)$ random draw from the posterior of vec($\mathbf{\Sigma}$).

24 RND_REGS

- **Purpose**
  Same as the rnd_regs procedure, but with $\mathbf{\Sigma}$ given.

- **Format**
  beta = rnd_regs(Y,X,b0,invBB0,Sigma)

- **Input**

Y	$(k \times n)$ observation matrix with the n observations of the k variates.
X	$(m \times n)$ observation matrix with the n observations of the m covariates.
b0	$(km \times 1)$ vector of prior means for vec($\mathbf{B}$).
invBB0	$(km \times km)$ inverse of the prior covariance matrix for vec($\mathbf{B}$).

- **Output**

beta	$(km \times 1)$ random draw from the posterior of vec($\mathbf{B}$).

25 ST_SAMPL

- **Purpose**
  In a Markov switching framework, it generate a state vector (discrete number in $\{1, 2, \ldots, r\}$), given the transition matrix P and the matrix of Hamilton-filtered probabilities.

- **Format**
  `s = st_sampl(xi,P)`

- **Input**

`xi`	$(m \times n)$ matrix of Hamilton-filtered probabilities for m states and n observations.
`P`	$(m \times m)$ transition matrix (columns sum to one) of the m states Markov chains.

- **Output**

`s`	$(n \times 1)$ vector of sampled states.

26 KALMANF

- **Purpose**
  It runs the Kalman filter on the dynamic linear model

$$\begin{aligned} \mathbf{y}_t &= \mathbf{Z}_t\boldsymbol{\alpha}_t + \mathbf{d}_t + \boldsymbol{\varepsilon}_t \\ \boldsymbol{\alpha}_t &= \mathbf{T}_t\boldsymbol{\alpha}_{t-1} + \mathbf{c}_t + \mathbf{R}_t\boldsymbol{\eta}_t, \end{aligned}$$

  with $\mathrm{E}(\boldsymbol{\alpha}_0) = \mathbf{a}_0$ and $\mathrm{Var}(\boldsymbol{\alpha}_0) = \mathbf{P}_0$.

- **Format**
  `{a,P,aa,PP,fe,F} = kalmanf(y,Z,d,H,T,c,R,Q,a0,P0)`

- **Input**

`y`	$(k \times n)$ matrix of k time series with n observation.
`Z`	$(k \times mn)$ matrix of n regression $(k \times m)$ matrices (one for each $t \in \{1, \ldots, n\}$).
`d`	$(k \times n)$ matrix of n $(k \times 1)$ vectors of constant terms (one for each $t \in \{1, \ldots, n\}$).
`H`	$(k \times kn)$ matrix of n covariance $(k \times k)$ matrices of the observational disturbances (one for each $t \in \{1, \ldots, n\}$).
`T`	$(m \times mn)$ matrix of n transition $(m \times m)$ matrices (one for each $t \in \{1, \ldots, n\}$).
`c`	$(m \times n)$ matrix of n $(k \times 1)$ vectors of constant terms (one for each $t \in \{1, \ldots, n\}$).
`R`	$(m \times gn)$ matrix of n $(m \times g)$ transformation matrices for the random vector $\boldsymbol{\eta}_t$ (one for each $t \in \{1, \ldots, n\}$).
`Q`	$(g \times gn)$ matrix of n $g \times g$ covariance matrices of the transition disturbance (one for each $t \in \{1, \ldots, n\}$).
`a0`	$(m \times 1)$ expectation vector for the state $\boldsymbol{\alpha}_0$.
`P0`	$(m \times m)$ covariance matrix for the state $\boldsymbol{\alpha}_0$.

- **Output**

`a`	$(m \times n)$ expectation vectors of the filtered states.
`P`	$(m \times mn)$ covariance matrices of the filtered states.
`aa`	$(m \times n)$ expectation vectors of the (1-step-ahead) predicted states.
`PP`	$(m \times mn)$ covariance matrices of the (1-step-ahead) predicted states.
`fe`	$(m \times n)$ (1-step-ahead) prediction error.
`F`	$(m \times mn)$ covariance matrices of the (1-step-ahead) prediction error.

27 SSAMPLER

- **Purpose**
  It generates an n-variate random draw from the joint smoothing distribution of the normal dynamic linear model

$$\begin{aligned} \mathbf{y}_t &= \mathbf{Z}_t\boldsymbol{\alpha}_t + \mathbf{d}_t + \boldsymbol{\varepsilon}_t \\ \boldsymbol{\alpha}_t &= \mathbf{T}_t\boldsymbol{\alpha}_{t-1} + \mathbf{c}_t + \mathbf{R}_t\boldsymbol{\eta}_t, \end{aligned}$$

  with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{H}_t)$, $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t\mathbf{Q}_t\mathbf{R}_t')$ and $\boldsymbol{\alpha}_0 \sim \mathcal{N}(\mathbf{a}_0, \mathbf{P}_0)$.

- **Format**
  `alpha = ssampler(a,P,T,Q,c,R)`

- **Input**

`a`	$(m \times n)$ matrix of n expectation vectors (one for each $t \in \{1, \ldots, n\}$) of the (Kalman) filtered states.
`P`	$(m \times mn)$ matrix of n covariance matrices (one for each $t \in \{1, \ldots, n\}$) of the (Kalman) filtered states.
`T`	$(m \times mn)$ matrix of n transition $(m \times m)$ matrices (one for each $t \in \{1, \ldots, n\}$).
`Q`	$(g \times gn)$ matrix of n $g \times g$ covariance matrices of the transition disturbance (one for each $t \in \{1, \ldots, n\}$).
`c`	$(m \times n)$ matrix of n $(k \times 1)$ vectors of constant terms (one for each $t \in \{1, \ldots, n\}$).
`R`	$(m \times gn)$ matrix of n $(m \times g)$ transformation matrices for the random vector $\boldsymbol{\eta}_t$ (one for each $t \in \{1, \ldots, n\}$).

- **Output**

`alpha`	$(m \times n)$ random draws from the joint smoothing distribution of the normal dynamic linear model.

28 SV_HSAMP

- **Purpose**
  It generates a random vector from the (approximated) conditional joint smoothing distribution of the variances h_t of the following stochastic volatility model

$$\begin{aligned} \log z_t^2 &= h_t + \log \varepsilon_t^2 \\ h_t &= \nu + \phi h_{t-1} + \eta_t, \end{aligned}$$

  with $\varepsilon \sim \mathcal{N}(0,1)$ and $\eta \sim \mathcal{N}(0,\sigma^2)$. The procedure is thought to be used in a Gibbs sampler, and therefore previously generated values of ν, ϕ, σ^2 and $\{h_t\}_{t=1}^T$ must be provided.

- **Format**
  `h = sv_hsamp(y,h,nu,phi,sigma2,m0,V0)`

- **Input**

`y`	$(n \times 1)$ log squared observations: $y = \log z^2$.
`h`	$(n \times 1)$ vector of previously sampled state variables.
`nu`	(1×1) constant of the AR(1) model for h_t.
`phi`	(1×1) autoregressive parameter of the AR(1) model for h_t.
`sigma2`	(1×1) variance of the innovations of the AR(1) model for h_t.
`m0`	(1×1) mean of the initial state h_0.
`V0`	(1×1) variance of the initial state h_0.

- **Output**

`h`	$(n \times 1)$ vector of sampled state variables from the joint smoothing distribution of h_t, $t = 1, \ldots, n$.

29 SV_LL

- **Purpose**
  It calculates the value of the (quasi) log-likelihood of the following stochastic volatility model:

$$\begin{aligned} \log z_t^2 &= h_t + \log \varepsilon_t^2 \\ h_t &= \nu + \phi h_{t-1} + \eta_t, \end{aligned}$$

  with $\varepsilon \sim \mathcal{N}(0,1)$ and $\eta \sim \mathcal{N}(0,\sigma^2)$.

- **Format**
  `ll = sv_ll(nps,y)`

- **Input**

`nps`	(3×1) vector containing the three parameters of the stochastic volatility model, respectively, ν, ϕ and σ.
`y`	$(n \times 1)$ log squared observations: $y = \log z^2$.

- **Output**

`ll`	(1×1) value of the (quasi) log-likelihood.